



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



Aplicación para comunicaciones en red

3CM17

Practica 01

Aplicación de archivos

Integrantes:

- ♥ Bocanegra Heziquio Yestlanezi
- ♥ Martinez Cruz José Antonio

Profesor: Moreno Cervantes Axel Ernesto

Índice

Índice de imágenes	3
Objetivo	4
Introducción	5
Sockets	5
Cliente – Servidor	5
Desarrollo	6
Interfaz principal	6
Subir.....	7
Descargar	8
Eliminar.....	9
Conclusiones.....	10
Bocanegra Heziquio Yestlanezi.....	10
Martínez Cruz José Antonio	10
Referencias	11

Índice de imágenes

Imagen 1 Interfaz principal	6
Imagen 2 Interfaz Subir.....	7
Imagen 3 Archivo/Carpeta subido correctamente.....	7
Imagen 4 botón descargar.....	8
Imagen 5 Botón eliminar.....	9

Objetivo

Realizar una aplicación programada en java utilizando sockets que permita al cliente ver carpetas y archivos local y remotamente, crear y eliminar carpetas local y remotamente, subir archivos o carpetas local y remotamente, descargar archivos y carpetas desde la carpeta remota, así como permitirle cambiar el nombre a un archivo o carpeta.

Introducción

Sockets

Un socket es un término que se refiere a un punto final de un canal de comunicación bidireccional entre dos programas que se ejecutan en una red [1].

un socket es una abstracción que permite que diferentes procesos en diferentes computadoras se comuniquen entre sí a través de una red utilizando un conjunto estándar de protocolos de comunicación. Los sockets se identifican mediante una combinación única de una dirección IP y un número de puerto [1].

Un programa que desea enviar o recibir datos a través de la red crea un socket y establece una conexión con otro socket en la red. Los datos se pueden enviar y recibir a través de los sockets utilizando una variedad de protocolos de comunicación, como TCP o UDP [2].

Los sockets son fundamentales en la programación de redes y se utilizan en una amplia variedad de aplicaciones, como servidores web, correo electrónico, transferencia de archivos y juegos en línea [2].

Cliente – Servidor

es un paradigma de comunicación en el cual un programa cliente solicita un servicio o recurso a otro programa servidor. El servidor es responsable de proporcionar el servicio o recurso solicitado y el cliente es responsable de enviar la solicitud y recibir la respuesta del servidor [3].

El modelo cliente-servidor se utiliza en una amplia variedad de aplicaciones de red, como servidores web, servidores de correo electrónico, bases de datos y juegos en línea. Algunos ejemplos comunes de cómo se usa el modelo cliente-servidor en redes son:

- ♥ En una aplicación de correo electrónico, el programa cliente solicita la descarga de nuevos correos electrónicos desde el servidor de correo. El servidor responde con los nuevos correos electrónicos y el cliente los muestra al usuario [3].
- ♥ En una aplicación de servidor web, el programa cliente (un navegador web) solicita una página web específica al servidor web. El servidor responde con la página web solicitada y el cliente la muestra al usuario [4].
- ♥ En una aplicación de juego en línea, el programa cliente solicita conectarse a un servidor de juegos. El servidor responde con una lista de juegos disponibles y el cliente elige uno para jugar [4].

Desarrollo

Interfaz principal

Como podemos observar en la imagen 1 tenemos la interfaz principal en java de la aplicación donde podemos encontrar del lado derecho los botones para Inicio/Actualizar archivos o carpetas, Subir archivos o carpetas, Descargar archivos y carpetas, así como el botón de Eliminar carpetas.

Del lado izquierdo se mostrarán los archivos o carpetas cargadas al presionar el botón Inicio/Actualizar.

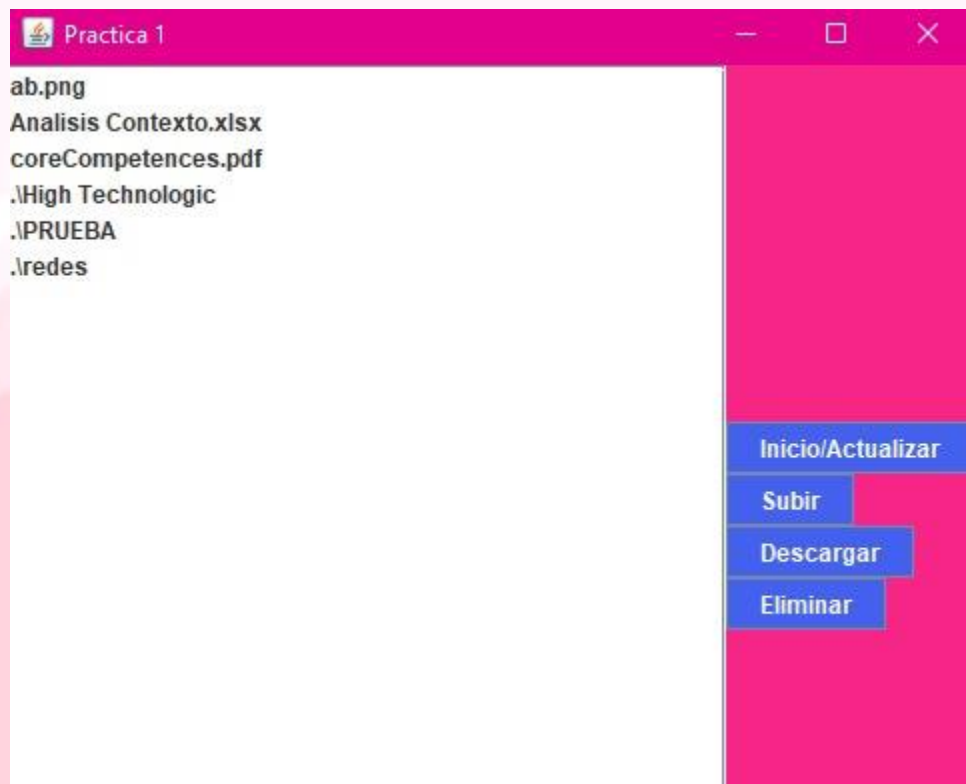


Imagen 1 Interfaz principal

Subir

En la imagen 2 podemos observar que al presionar el botón de subir se desplegará la siguiente ventana, donde nos permitirá seleccionar un archivo o carpeta para cargarla al presionar open.

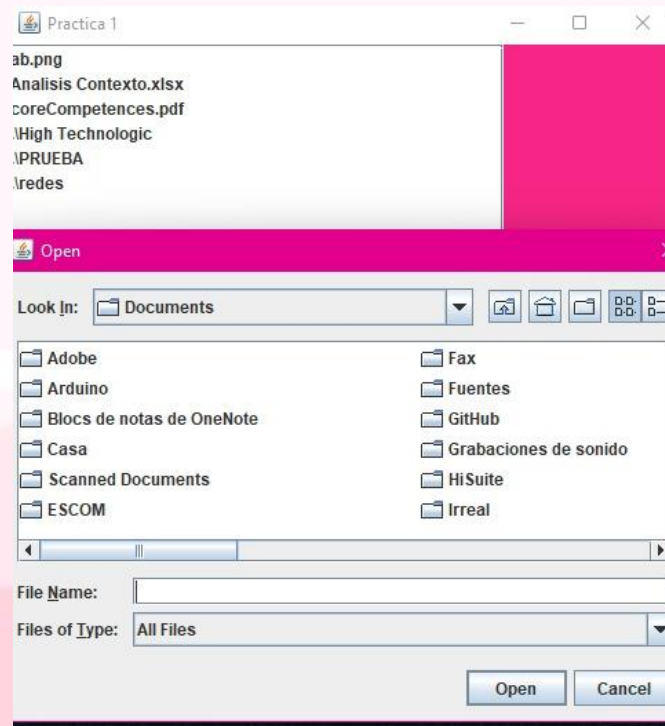


Imagen 2 Interfaz Subir

Después nos arrojará el siguiente mensaje como se muestra en la imagen 3 que nos indica que el archivo o carpeta se subió adecuadamente y ya podremos visualizarla en la interfaz principal.

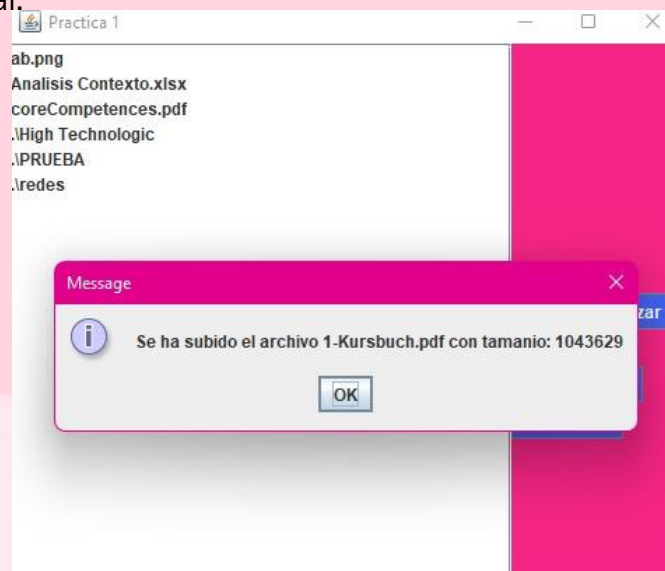


Imagen 3 Archivo/Carpeta subido correctamente

Descargar

En la imagen 4 podemos visualizar el funcionamiento correcto del botón descargar.

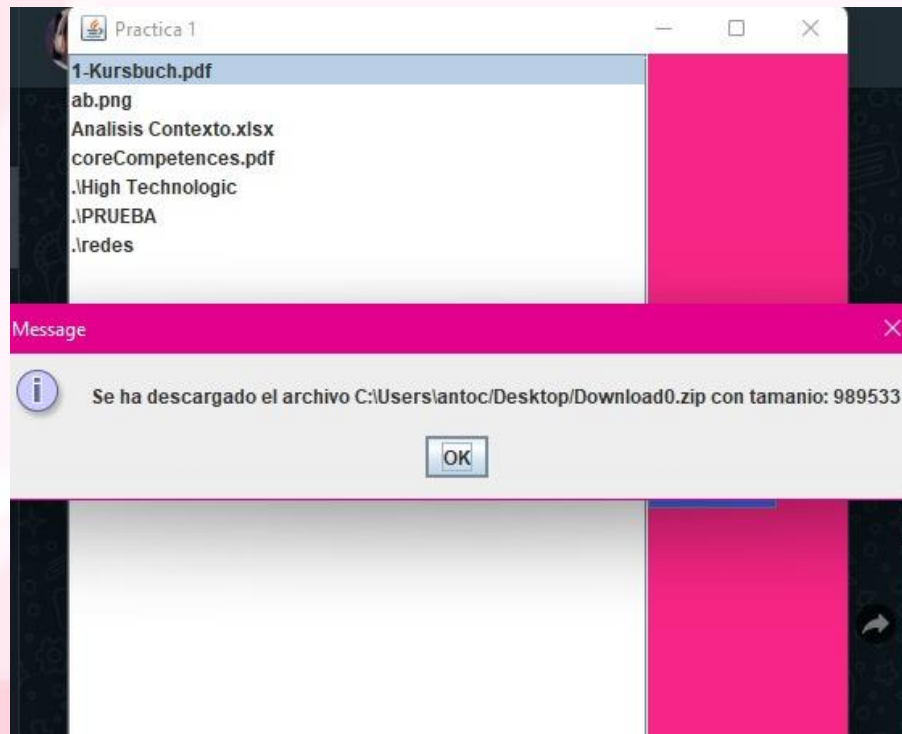


Imagen 4 botón descargar

Al seleccionar las carpetas o archivos que se encuentran en la interfaz principal, cuando seleccionamos el botón de descargar, nos arroja la leyenda que indica que la carpeta comprimida se a descargado correctamente y se encuentra en la ruta especificada en la imagen 4.

Eliminar

A continuación, y por último podemos observar en la imagen 5 el funcionamiento del botón eliminar, que te permitirá eliminar archivos o carpetas desde la interfaz principal.

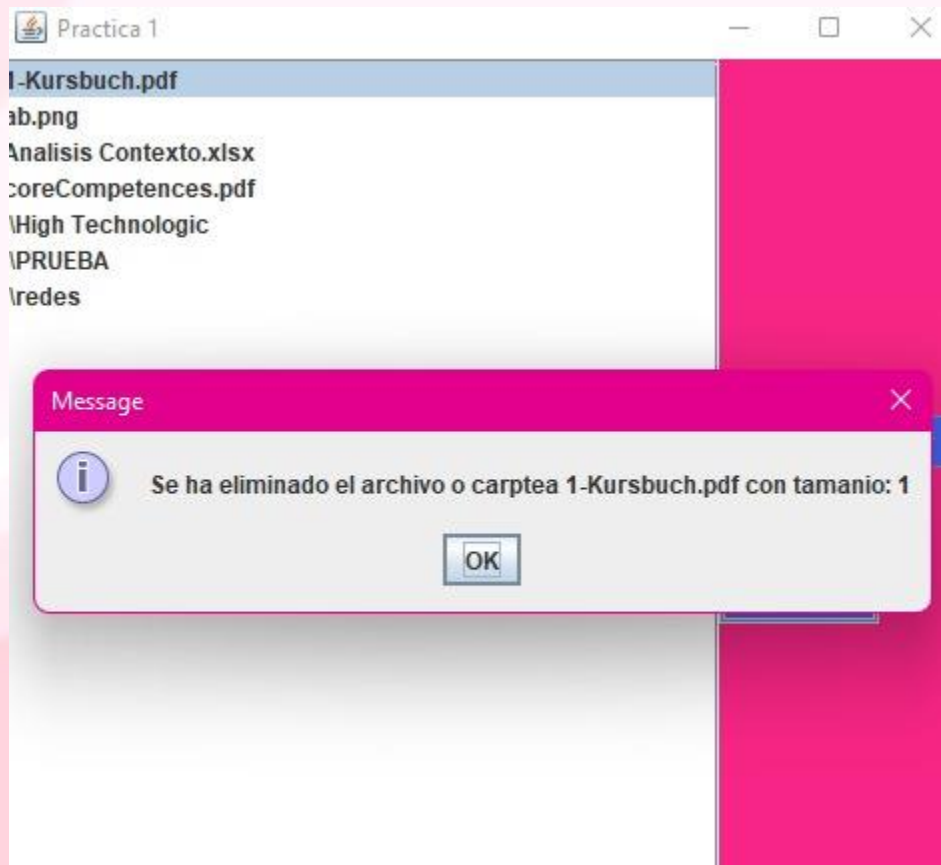


Imagen 5 Botón eliminar

Al momento de seleccionar un archivo o carpeta de la interfaz principal nos arrojará la leyenda de que el archivo a sido eliminado correctamente.

Conclusiones

Bocanegra Heziquio Yestlanezi

Puedo concluir que, la aplicación de archivos en Java utilizando sockets permite a los clientes conectarse a un servidor y realizar diversas operaciones en las carpetas y archivos almacenados tanto localmente como en el servidor. La aplicación permite visualizar, crear y eliminar carpetas, subir y descargar archivos o carpetas, así como también cambiar el nombre a un archivo o carpeta en la carpeta remota. Esto permite a los usuarios una gestión más eficiente y efectiva de sus archivos, ya que pueden realizar todas estas operaciones desde una única plataforma en línea. Además, esta aplicación es una excelente manera de aprender sobre la comunicación entre aplicaciones utilizando sockets y cómo se pueden aplicar en proyectos del mundo real.

Martínez Cruz José Antonio

En esta práctica se logró implementar de manera correcta el administrador de archivos, permitiendo tener una conexión exitosa entre el servidor y el cliente. A pesar de que existieron inconvenientes como la eliminación de varios archivos y de carpetas en carpetas, estos detalles lograron ser llevados gracias a la creación de listas de elementos segregadas, permitiendo la manipulación de diversos archivos. Fuera de esta situación, no hubo detalles con la codificación de los sockets utilizados en esta práctica.

Referencias

- [1] S. Keshav, "An engineering approach to computer networking: ATM networks, the internet, and the telephone network," 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997, pp. 64-65.
- [2] M. Tanenbaum, "Computer networks," 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003, pp. 128-132.
- [3] R. Stevens, "TCP/IP Illustrated, Volume 1: The Protocols," 2nd ed. Boston, MA, USA: Addison-Wesley Professional, 2011, pp. 3-5.
- [4] A. S. Tanenbaum and D. J. Wetherall, "Computer Networks," 5th ed. Boston, MA, USA: Pearson, 2011, pp. 107-111.